Experiments on Universal Rigidity of Bipartite Graphs

The Harvard community has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Accessed</td>
<td>October 18, 2017 5:54:49 AM EDT</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:14398546">http://nrs.harvard.edu/urn-3:HUL.InstRepos:14398546</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at <a href="http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA">http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA</a></td>
</tr>
</tbody>
</table>

(Article begins on next page)
Experiments on Universal Rigidity of Bipartite Graphs

Deborah Barbosa Alves

Department of Computer Science
In partial fulfillment of the requirements for the joint degree of A.B. in Computer Science and Mathematics

Harvard College
April 1st, 2015
Abstract

Our goal is to characterize necessary and sufficient conditions for the universal rigidity of bipartite frameworks. Previous work describe ways to test universal rigidity through semidefinite programming using stress matrix as a tool. Also, it had been shown a relationship between rigidity of a bipartite framework and quadric separability of the two sets of vertices. In particular, previous work showed that given a complete bipartite framework, separability by a quadric implied non-rigidity of the framework. Based on this, a reasonable conjecture was that the reciprocal could also be true. Our goal was to develop experiments using semidefinite programming to validate this conjecture for complete bipartite framework, and observe the behavior of rigidity for incomplete bipartite frameworks.
Acknowledgements

I would like to immensely thank my advisor, Prof. Steven J. Gortler, for introducing me to this interesting topic, for the excitement when discussing new results, and all the patience when explaining everything to me.

I would also like to thank my family and friends for the great support and encouragement that pushed me to my academic achievements.
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>5</td>
</tr>
<tr>
<td>1 Background</td>
<td>7</td>
</tr>
<tr>
<td>1.1 Basic Definitions</td>
<td>7</td>
</tr>
<tr>
<td>1.2 Previous Results on Rigidity</td>
<td>9</td>
</tr>
<tr>
<td>1.3 Trilateration</td>
<td>10</td>
</tr>
<tr>
<td>2 Bipartite Frameworks</td>
<td>14</td>
</tr>
<tr>
<td>2.1 Separation criterion</td>
<td>15</td>
</tr>
<tr>
<td>2.1.1 From conic to hyperplane separability</td>
<td>15</td>
</tr>
<tr>
<td>2.1.2 Radon theorem</td>
<td>16</td>
</tr>
<tr>
<td>2.2 Previous results</td>
<td>18</td>
</tr>
<tr>
<td>2.3 Certificates</td>
<td>19</td>
</tr>
<tr>
<td>2.4 Particular cases</td>
<td>19</td>
</tr>
<tr>
<td>3 Experiments</td>
<td>21</td>
</tr>
<tr>
<td>3.1 Semidefinite Programming</td>
<td>21</td>
</tr>
<tr>
<td>3.2 Experiment setup</td>
<td>23</td>
</tr>
<tr>
<td>3.2.1 Generating Points</td>
<td>23</td>
</tr>
<tr>
<td>3.2.2 SDP Inputs</td>
<td>24</td>
</tr>
<tr>
<td>3.3 Test Cases and Results</td>
<td>25</td>
</tr>
</tbody>
</table>
3.4 Tests on Incomplete Bipartite Graphs
Introduction

A framework \((G,\mathbf{p})\) is a graph \(G\) along with an embedding of its vertices which are mapped to points \(\mathbf{p} = (\mathbf{p}_1,\ldots,\mathbf{p}_N)\) in \(\mathbb{R}^d\). We say that a framework is \textit{universally rigid} if any other embedding of its graph that preserves edge lengths is an euclidean transformation of the original embedding. Our goal is to better characterize the conditions of universal rigidity on bipartite frameworks (frameworks whose graph is bipartite) by finding a computable criterion that determines whether or not the bipartite framework is rigid.

On chapter 1, we introduce the concept of stresses and stress matrices of a framework, which is an assignment of a real value to each edge of the graph, and will be our main tool to characterize rigidity. We go over previous results that relate the existence of equilibrium stress matrices with the rigidity of general frameworks (not necessarily bipartite). These results will be the basic facts used to test and prove the computable criterion for bipartite frameworks that we’ll explore on the following chapters.

On chapter 2, we focus on previous findings particular to complete bipartite graphs. First, we go over the concept of quadric separability of the two sets vertices of a bipartite graph, which is the main computable property that is related to the rigidity of bipartite frameworks. We conclude that quadric separability in our framework space is equivalent to hyperplane separability in a higher dimensional space, which, by the hyperplane separation theorem, is equivalent to non intersection of two convex hulls. That is, to analyze quadric separability of the two sets of vertices, it is enough to analyze whether or not the convex hull of the two sets of points intersect. Thus, we reduce the separability condition to a linear system, which will be used on our experiments. Finally, we
list previous results that tie the quadric separability condition with rigidity of complete bipartite frameworks and a conjecture based on those, which will be the main topic of our experiments.

On chapter 3 we go over our experiment process. First, we describe semidefinite programming and how it relates to the rigidity test, because the main theorems of rigidity described in chapter 1 are based on existence of positive semidefinite stress matrices. Then we describe the algorithm to test rigidity of complete bipartite frameworks to support our conjecture, the generation of cases, setup of the semidefinite program, and the analysis of the results, which give a strong motivation to the correctness of the conjecture. Finally, we go over tests on incomplete bipartite graphs, and the results on the rigidity of those.
Chapter 1

Background

1.1 Basic Definitions

Given a graph $G = (V, E)$ of $N$ vertices, we define a framework $(G, p)$ as the graph $G$ along with a configuration $p = (p_1, \ldots, p_N)$ of points $p_i$ in $\mathbb{R}^d$ such that the points of $p$ have a $d$-dimensional affine space. The points $p$ represent an embedding of the vertices $V$ in $\mathbb{R}^d$, that is, each vertex $v_i$ in $G$ is associated with a point $p_i$. In many cases, we’ll denote each $p_i$ as the point in $\mathbb{R}^d$ in vector form, and then $p$ will be a $d \times N$ matrix. We can then interpret the framework as a structure on $\mathbb{R}^d$ where each vertex represents an articulation point in space, and each edge represents a rigid bar connecting two articulation points.

With that interpretation of rigid bars, we define the concept of rigidity of a framework. Intuitively, a framework is rigid if its structure cannot be flexed; for instance, a square is not rigid, since it can be flexed to a rhombus (or even a line segment), while any triangle is rigid. That is, a framework is rigid if there are no other embeddings of the same graph $G$ preserving the lengths of edges in the framework, other than congruent configurations of the same framework. More formally, we define universal rigidity as follows:

**Definition 1.1.1 (Universal Rigidity).** Given a framework $(G, p)$ on $\mathbb{R}^d$, we say that it is universally rigid if any other configuration $q = (q_1, \ldots, q_N)$ of points in $\mathbb{R}^D$, with $D \geq d$, is such that
\[ |p_i - p_j| = |q_i - q_j| \text{ for every edge } \{i, j\} \text{ of } G, \text{ then } p \text{ and } q \text{ are congruent, that is, } |p_i - p_j| = |q_i - q_j| \text{ for every } 1 \leq i, j \leq N. \]

A related concept is global rigidity, which satisfies the same definition except we restrict the configuration \( q \) to \( \mathbb{R}^d \). Clearly, if a framework is universally rigid, then it is also globally rigid.

Our goal is to understand the criteria for universal rigidity of frameworks and, in particular, frameworks on bipartite graphs. To that purpose, we’ll use stresses as our main tool.

**Definition 1.1.2 (Equilibrium Stress).** A stress \( \omega \) of a framework is an assignment of a real scalar \( \omega_{ij} = \omega_{ji} \) to each edge \( \{i, j\} \) of the framework graph \( G \). We assume \( \omega_{ij} = 0 \) when \( \{i, j\} \) is not an edge of \( G \). We say that a stress \( \omega \) is in equilibrium if the equation

\[
\sum_j \omega_{ij} (p_i - p_j) = 0
\]

holds for all vertices \( i \) of \( G \).

For each stress, we associate a matrix that we call stress matrix which will help us better explore the properties of equilibrium stresses.

**Definition 1.1.3 (Stress Matrices).** Given a stress \( \omega \) of a framework, we define an associated matrix \( \Omega \) as a \( N \times N \) matrix, where \( N \) is the number of vertices in the framework, such that the \((i, j)\) entry of \( \Omega \) is \(-\omega_{ij}\) for all \( i \neq j \), and the diagonal entries \( \Omega_{k,k} = \lambda_k \) are defined so that the sum of all rows and columns are zero. That is, \( \lambda_k = \sum_{i=1, i \neq k}^N \omega_{ik} \). Also note, in particular, that \( \Omega \) is a symmetric matrix.

We say that \( \Omega \) is an equilibrium stress matrix for \((G, p)\) if its associated stress is an equilibrium stress. Note that the equilibrium condition in 1.1 can be rewritten as

\[
\sum_j \omega_{ij} (p_i - p_j) = (\sum_j \omega_{i,j}) p_i - \sum_j \omega_{ij} p_j = \lambda_i p_i + \sum_{j \neq i} (-\omega_{ij}) p_j = 0
\]
and therefore $\Omega$ is an equilibrium stress matrix if and only if

\[
\Omega p^T = \begin{pmatrix}
\lambda_1 & -\omega_{1,2} & \ldots & -\omega_{1,N} \\
& \ddots & & \\
-\omega_{N,1} & -\omega_{N,2} & \ldots & \lambda_N
\end{pmatrix}
\begin{pmatrix}
p_1 \\
\vdots \\
p_N
\end{pmatrix}
= \begin{pmatrix}
\lambda_1 p_1 + \sum_{j \neq 1} (-\omega_{1j}) p_j \\
\vdots \\
\lambda_N p_N + \sum_{j \neq N} (-\omega_{Nj}) p_j
\end{pmatrix}
= 0.
\]

(1.2)

We’ll use equilibrium stress matrices as our main tool to find sufficient and necessary conditions for rigidity in frameworks. The following section will list the main theorems we will be using in the experiments for bipartite frameworks.

### 1.2 Previous Results on Rigidity

The main results in this section are strongly linked to equilibrium stress matrices. In particular, the results rely on the existence of *positive semidefinite* matrices, defined below.

**Definition 1.2.1** (Positive Semidefinite (PSD) Matrix). A real $n \times n$ matrix $M$ is called positive semidefinite if $x^T M x \geq 0$ for every $x \in \mathbb{R}^n$.

There are also other ways to define a positive semidefinite matrix. The following properties are equivalent to $M$ being positive semidefinite:

(a) All eigenvalues of $M$ are non negative

(b) $M$ an be written as $M = A^T A$ for some matrix $P$.

The PSD property will be an important condition on the stress matrix in the following theorems. But before we describe our first fundamental theorem, we need one more definition.

**Definition 1.2.2.** We say that a collection $v = \{v_1, \ldots, v_n\}$ of non-zero vectors in $\mathbb{R}^d$ lie on a conic at infinity if there is a symmetric $d \times d$ matrix $A$ such that for all $i = 1, \ldots, m$, $v_i^T A v_i = 0$. That is, if the points all lie on a conic when regarded as points in the real projective space $\mathbb{R}P^{d-1}$.

Given an equilibrium stress matrix for $(G, p)$, we know from equation 1.2 that $\Omega p^T = 0$. Also, from the definition of stress matrix, we have that $\Omega \mathbf{1} = \mathbf{0}$, where $\mathbf{1}$ is the $N$-dimensional vector of
all 1s. Therefore, if the affine span of $p$ has dimension $d$, the rows of $p$ along with 1 are linearly independent, and thus the kernel of $\Omega$ has dimension at least $d + 1$. Thus, the rank of $\Omega$ is at most $N - d - 1$.

The following theorem from [1] provide a sufficient condition for universal rigidity, which will be fundamental for our experiments.

**Theorem 1.2.3.** Let $(G, p)$, with $p = (p_1, \ldots, p_N)$, be a framework on $\mathbb{R}^d$ whose affine span of $p$ is all of $\mathbb{R}^d$, and with an equilibrium stress matrix $\Omega$. Suppose further

i. $\Omega$ is positive semidefinite (PSD)

ii. The rank of $\Omega$ is $N - d - 1$.

iii. The vectors $p_1, \ldots, p_N$ do not lie on a conic at infinity.

Then $(G, p)$ is universally rigid.

In other words, if the vertices do not lie on a conic at infinity, and if the framework has a PSD equilibrium matrix of maximum rank, then it is universally rigid.

Also, the following theorem from [7] states that the conditions of the theorem above are also necessary.

**Theorem 1.2.4.** If $(G, p)$ is a universally rigid framework with $N$ vertices whose affine span is $d$-dimensional, $d \leq N - 2$, then it has a non-zero equilibrium stress with a PSD matrix $\Omega$.

Therefore for any universally rigid framework, there is a PSD equilibrium stress matrix that serves as a certificate for its rigidity. We’ll use this result on our experiments on chapter 3 by looking for PSD equilibrium stress matrices to prove the rigidity of frameworks.

### 1.3 Trilateration

To prove the universal rigidity of a framework $(G, p)$ in $\mathbb{R}^d$, sometimes it will be enough to prove the rigidity of a subframework. Consider a vertex $p$ in the framework with degree at least $d + 1$. 
Now suppose the remaining $N - 1$ vertices form a universally rigid subframework, and because of that, suppose we fix the positions of the $N - 1$ vertices. To prove the universal rigidity of the entire framework, it is then enough to show that given the fixed positions of the $N - 1$ vertices, the point $p$ must have a fixed position as well to preserve the edge lengths.

So let $p_1, \ldots, p_{d+1}$ be the neighboring points of $p$, and $d_i = |p - p_i|$. We want to know the number of solutions to the system of $(d + 1)$ equations

$$ |x - p_i| = d_i \Rightarrow x^2 + p_i^2 - 2x \cdot p_i = d_i^2 $$

where $x$ and $p_i$ can also be taken in a higher dimension (we can pad $p_i$s with 0s). We can rearrange the equations by subtracting the last equation from the others and obtain linear equations:

$$ \begin{align*}
    p_1^2 - p_{d+1}^2 - 2x \cdot (p_1 - p_{d+1}) &= d_1^2 - d_{d+1}^2 \\
    \vdots \\
    p_d^2 - p_{d+1}^2 - 2x \cdot (p_d - p_{d+1}) &= d_d^2 - d_{d+1}^2 \\
    x^2 + p_{d+1}^2 - 2x \cdot p_{d+1} &= d_{d+1}^2
\end{align*} $$

$$ \Rightarrow \begin{pmatrix}
    (p_1 - p_{d+1})^T \\
    \vdots \\
    (p_d - p_{d+1})^T \\
    x^2 + p_{d+1}^2 - 2x \cdot p_{d+1}
\end{pmatrix}
\begin{pmatrix}
    d_1^2 - d_{d+1}^2 - (p_1^2 - p_{d+1}^2) \\
    \vdots \\
    d_d^2 - d_{d+1}^2 - (p_d^2 - p_{d+1}^2) \\
    d_{d+1}^2
\end{pmatrix}
\Rightarrow Ax = b $$

The matrix $A$ has rank $d$, since we’re assuming the $d$ points are in general linear position. Therefore, the linear system $Ax = b$ assumes only one solution, which is $x = p$. Therefore, the vertex $p$ has a fixed position, so the framework will be rigid as desired.

This is a generalization of the common trilateration argument in the plane of the 3D space. Given 3 circles in the plane whose centers are not collinear (that is, centers in general position), the three circles intersect in at most one point. In general, given $d + 1$ hypersphere in $\mathbb{R}^d$ with centers
in general linear position, they intersect in at most one point. Figure 1.1 illustrates the trilateration in the plane.

![Figure 1.1: Illustration of the trilateration argument on the plane.](image)

Note, moreover, that if a point has degree less than $d + 1$, in general the framework won’t be rigid, since even if the remaining subframework is rigid, this point can move around. This is because the linear system above will have a solution space with dimension at least one, and the quadratic equation, which can be extended to a higher dimension, will intersect this solution space in more than one point. For instance, in the plane, we have that if a vertex has degree at most 2, it can vary its position along a circle in $\mathbb{R}^3$, as illustrated in figure 1.2.

![Figure 1.2](image)
In conclusion, we have that if a framework in $\mathbb{R}^d$ has a vertex with degree less than $d + 1$, and its points are in general linear position, the framework cannot be rigid. Moreover, if a vertex has degree at least $d + 1$ and the subframework with this vertex removed is universally rigid, then the framework will be universally rigid.
Chapter 2

Bipartite Frameworks

We will focus on finding the rigidity conditions for bipartite frameworks, that is, frameworks whose graph $G$ is bipartite, and in particular, complete bipartite. So let’s denote now by $p = (p_1, \ldots, p_n)$ and $q = (q_1, \ldots, q_m)$ the two sets of points in our bipartite framework, so that we have $N = n + m$ vertices total. Denote by $K(n, m)$ the complete bipartite graph with $m + n$ vertices, so we’ll denote a complete bipartite framework as $(K(n, m), (p, q))$. Assume also that the points $p_i$ and $q_i$ denote $d \times 1$ vectors, representing the points in $\mathbb{R}^d$.

The condition of rigidity we will focus on is related to whether or not the points $p$ and $q$ can be separated by a quadric surface in $\mathbb{R}^d$. To correctly define this, we will adapt the points in $\mathbb{R}^d$ to points in the projective space $\mathbb{RP}^d$ by adding a coordinate 1 to the end of each point. That is, for every vector $x \in \mathbb{R}^d$, define $\hat{x} \in \mathbb{R}^{d+1}$ as the vector $\hat{x} = \begin{bmatrix} x \\ 1 \end{bmatrix}$.

Definition 2.0.1 (Quadric Separability). We say that the points $p = (p_1, \ldots, p_n)$ and $q = (q_1, \ldots, q_m)$ in $\mathbb{R}^d$ are strictly separated by a quadric if there is a $(d + 1) \times (d + 1)$ symmetric matrix $A$ such that

$$\hat{p}_i^T A \hat{p}_i > 0 \quad \text{and} \quad \hat{q}_j^T A \hat{q}_j < 0.$$  \quad (2.1)

for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$. 

The quadric separability condition can actually be simplified to a linear condition, as we will see in the following section.

2.1 Separation criterion

Note that, given a vector $\mathbf{x} \in \mathbb{R}^d$ and a symmetric $(d+1) \times (d+1)$ matrix $A$, we have from basic trace properties

$$\hat{\mathbf{x}}^T A \hat{\mathbf{x}} = \text{tr}(\hat{\mathbf{x}}^T A \hat{\mathbf{x}}) = \text{tr}(A \hat{\mathbf{x}} \hat{\mathbf{x}}^T) = A \cdot (\hat{\mathbf{x}} \hat{\mathbf{x}}^T) \quad (2.2)$$

where $\cdot$ in the last expression represents the entry-wise product between the $(d+1) \times (d+1)$ matrices $A$ and $\hat{\mathbf{x}} \hat{\mathbf{x}}^T$. Based on this, it makes sense to take a further look at the matrices $\hat{\mathbf{x}} \hat{\mathbf{x}}^T$.

2.1.1 From conic to hyperplane separability

Let $S_d$ be the set of all symmetric $(d+1) \times (d+1)$ matrices with last entry (entry $(d+1,d+1)$) equal to 1. Note that this set forms a matrix space of dimension $(d+1)(d+2)/2 - 1$, since our degrees of freedom correspond to the entries in the upper triangular matrix, except the last one.

**Definition 2.1.1 (Veronese map).** We define the map $\mathcal{V} : \mathbb{R} \to S_d$ as

$$\mathcal{V} : \mathbf{x} \mapsto \hat{\mathbf{x}} \hat{\mathbf{x}}^T.$$  

*Note that it is well defined since, by the definition of $S_d$, $\hat{\mathbf{x}} \hat{\mathbf{x}}^T \in S_d$.*

Note that from 2.1 and 2.2 the points $\mathbf{p}$ and $\mathbf{q}$ are strictly separated by a quadric, if and only if there is a symmetric matrix $A$ such that

$$A \cdot \mathcal{V}(\mathbf{p}_i) > 0 \text{ and } A \cdot \mathcal{V}(\mathbf{q}_j) < 0 \quad (2.3)$$

for all $i = 1, \ldots, n$ and $j = 1, \ldots, m$.

Consider now the bijection $f : S_d \to \mathbb{R}^{(d+1)(d+2)/2-1}$ which takes the symmetric matrix $A \in S_d$, and returns the vector in $\mathbb{R}^{(d+1)(d+2)/2-1}$ which corresponds to the "vectorization" of the
upper triangular entries in $A$, except the last one (which is always 1). Finally, let $M : \mathbb{R}^d \to \mathbb{R}^{(d+1)(d+2)/2-1}$ be the map $M = f \circ V$.

Note that $A \cdot V(x)$ is just a linear combination of the entries of $x$ plus a constant (because of the last diagonal entry which is 1). That is, it can be rewritten as

$$a^T M(x) + b$$

where $a$ is a vector in $\mathbb{R}^{(d+1)(d+2)/2-1}$ and $b$ is a constant, where both only depend on entries of $A$. Therefore the condition in 2.3 is equivalent to saying that the points $(M(p_1), \ldots, M(p_n))$ and $(M(q_1), \ldots, M(q_m))$ are separated by a hyperplane in $\mathbb{R}^{(d+1)(d+2)/2-1}$.

### 2.1.2 Radon theorem

Note that, by the hyperplane separation theorem, two sets of points can be separated by a hyperplane if and only if their convex hulls do not intersect. The following theorem and its proof provide a deeper understanding of the convex hull intersection condition and, consequently, of the separability condition.

**Theorem 2.1.2 (Radon).** Any set of $N \geq D + 2$ points in $\mathbb{R}^D$ can be partitioned into two disjoint sets whose convex hulls intersect.

**Proof.** Let $S = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^D$ be our set of points, taken as vectors in $\mathbb{R}^D$. Consider the following linear system:

$$\sum_{i=1}^{N} t_i x_i = 0 \text{ and } \sum_{i=1}^{N} t_i = 0$$

$$\Rightarrow \begin{bmatrix} x_1 & \cdots & x_N \\ 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix} = X t = 0. \quad (2.4)$$

Since we have $D + 1$ equations for $N \geq D + 2$ unknowns, there must be a nonzero solution $t = (t_1, \ldots, t_N)$ for the system. In other words, since the domain has dimension $N \geq D + 2$, and
the rank of $X$ is at most $D + 1$, we must have a kernel of dimension at least 1. Therefore, let $A$ be the set of indices $i$ such that $t_i \geq 0$, and $B$ the analogous for $t_i < 0$. Then we have that $A$ and $B$ define a partition of the $D + 2$ points. Moreover, the convex hulls of both partitions intersect, since, from the linear system

$$\sum_{i=1}^{N} t_i = 0 \Rightarrow \sum_{i \in A} t_i = \sum_{i \in B} (-t_i) = T > 0$$

and

$$\sum_{i \in A} t_i x_i = \sum_{i \in B} -t_i x_i = P.$$

where, in the last equation, $P$ denotes the intersection point we found between both convex hulls. Note that since we’re assuming that $t$ is a nonzero solution, the sets $A$ and $B$ can’t be empty.

The proof of Radon theorem give us an algorithm to divide $N$ points in two sets that are not strictly separated by a conic. To do so, we can

1. Get the image $x' = M(x) \in \mathbb{R}^D$ of each point $x \in \mathbb{R}^d$, where $D = (d + 1)(d + 2)/2 - 1$.

2. Solve the linear system in 2.4

3. Separate the points based on the sign of the coefficients from the solution of the system.

The proof of Radon theorem also gives us some tools that we’ll use further in the experiments, noted on the following remarks:

**Remark 2.1.3.** Note that if we have that our points are in general linear position in $\mathbb{R}^D$ ($D = (d + 1)(d + 2)/2 - 1$), that is, if the columns of $X$ are linearly independent, we need at least $D + 2$ points to have a nonzero solution for 2.4, otherwise the kernel of $X$ would have to be 0. That is, if we have less than $D + 2$ points in general linear position in $\mathbb{R}^d$, any division of these points can be strictly separable by a conic. However, if they are not in general linear position, we might have a the convex hull intersection with less points. So in our problem we could consider, for instance, $D + 1$ points in a conic in $\mathbb{R}^d$, so that their image in $\mathbb{R}^D$ by $M$ will be in a hyperplane.
Remark 2.1.4. The coefficients we find in the linear system 2.4, will split the points in $\mathbb{R}^d$ into two groups $p = (p_1, \ldots, p_n)$ and $q = (q_1, \ldots, q_m)$ which are not strictly separable by a conic. From what we’ve seen above, this means that

$$
\sum_{i=1}^{n} \lambda_i M(p_i) = \sum_{i=1}^{m} \mu_i M(q_i) \Rightarrow \sum_{i=1}^{n} \lambda_i V(p_i) = \sum_{i=1}^{m} \mu_i V(q_i).
$$

We will call the coefficients $\lambda_1, \ldots, \lambda_n, \mu_1, \ldots, \mu_m$ the Radon coefficients of this point configuration.

2.2 Previous results

Our experiments that are described in the next chapter are based on previous results that related the conic separability with universal rigidity of the frameworks. The following theorems are the main results of rigidity focused on complete bipartite graphs that preceded the experiments.

The first theorem, proved in [2], describes a necessary condition for the universal rigidity of a complete bipartite framework, which was the main incentive to our experiments.

Theorem 2.2.1. If $(K(n,m), (p,q))$ is a bipartite framework in $\mathbb{R}^d$, with $m + n \geq d + 2$, such that the partition vertices $(p,q)$ are strictly separated by a quadric, then it is not universally rigid.

That is, this theorem relates quadric separability and rigidity. So it is reasonable to ask if the reciprocal is also true, that is, if strict quadric non separability implies rigidity. And this is main problem we’ll investigate in the experiments in the next chapter. More precisely, we want to test if the following conjecture is true:

Conjecture 2.2.2. If $(K(n,m), (p,q))$ is a complete bipartite framework in $\mathbb{R}^d$ such that the partition vertices $(p,q)$ are not strictly separated by a quadric and have affine span $\mathbb{R}^d$, then it is universally rigid.
2.3 Certificates

Suppose we have a complete bipartite framework \((K(n, m), (p, q))\) in \(\mathbb{R}^d\) such that the partition \((p, q)\) is not strictly separated by a quadric. Now consider a subframework \((K(n', m'), (p', q'))\), such that \(m', n' \geq d + 1\) and \((p', q')\) is also not strictly separated by a quadric. Then, because of the trilateration criterion covered in section 1.3, if this subframework is rigid, then so is \((K(n, m), (p, q))\).

And, if the affine span of the points is all \(\mathbb{R}^d\), in order to be non separable, each set of points must indeed satisfy \(m', n' \geq d + 1\), so we can always apply the trilateration.

Therefore, it is enough to analyze the smaller cases with \(N\) equal to the minimum number of points \(N_0\) required in the non separability criterion (as mentioned, in general \(N_0 = (d+1)(d+2)/2 + 1\)), since if \(N\) is greater, we can remove vertices until we have \(N_0\) which are still not separated.

2.4 Particular cases

We’ll start the investigation on conjecture 2.2.2 by analyzing some low dimensional cases. First, note that on bipartite frameworks the equilibrium stress matrix \(\Omega\) has the form

\[
\Omega = \begin{bmatrix}
A & -W \\
-W^T & B
\end{bmatrix}
\] (2.5)

where \(W\) is a \(n \times m\) matrix representing the edge stresses of the framework for the \(n \times m\) edges, and \(A\) and \(B\) are diagonal matrices, where the diagonal entries of \(A\) are the row sums of \(W\), and the diagonal entries of \(B\) are the column sums of \(W\) (so that the row and column sums of \(\Omega\) are 0).

The following theorem, which is an implication of Theorem 6 in [3], describes a relationship between stress matrices and the radon coefficients, which help us link the separability condition with the rigidity condition.

**Theorem 2.4.1.** Let \((K(n, m), (p, q))\) be a bipartite framework such that the affine span of \(p\) is all of \(\mathbb{R}^d\) and the affine span of \(q\) is all of \(\mathbb{R}^d\). Also, suppose \(p\) and \(q\) are not strictly separable by a quadric. Then there is an equilibrium stress matrix \(\Omega\) such that its diagonal consists of radon
coefficients for the point configuration \((p, q)\). In particular, there is a stress matrix with non negative diagonal.

That is, if \(\Omega\) above represents an equilibrium stress matrix for \((p_1, \ldots, p_n, q_1, \ldots, q_m)\) in this order, and it’s diagonal is \(a_1, \ldots, a_n, b_1, \ldots, b_m\), we have that

\[
\sum_{i=1}^{n} a_i \mathcal{V}(p_i) = \sum_{i=1}^{m} b_i \mathcal{V}(p_j).
\]

So suppose we have a framework \((K(n, m), (p, q))\) in \(\mathbb{R}^d\) which is not strictly separated by a quadric, and it is minimal so that \(n + m = N_0 = (d+1)(d+2)/2 + 1\). Then, from the theorem 2.4.1 above, it has an equilibrium stress matrix with positive diagonal. Moreover, since the stress matrix is of the form shown in 2.5, it will have a diagonal block of size \(m\). We know that \(n, m \geq d + 1\), so suppose \(n\) (or \(m\)) is exactly \(d+1\). Then, the matrix will have a diagonal block of size \(m = N - d - 1\), and therefore its rank is at least \(N - d - 1\), proving that this stress matrix has maximum rank. Therefore, by theorem 1.2.3, this will be universally rigid.

This fact helps us analyze special low dimensional cases. That is, we proved that if either \(p\) or \(q\) have size \(d + 1\), then the conjecture 2.2.2 holds. For \(d = 1\), we have \(N_0 = 4\), and we must have \(d + 1 = 2\) points in each side at least, so our only case is \(K(2, 2)\), and then the conjecture holds. For \(d = 2\), we have \(N_0 = 7\), and the only case is \(K(3, 4)\), which also satisfies the conjecture. For \(d = 3\), the possible cases are \(K(4, 7)\) and \(K(5, 6)\). For \(K(4, 7)\), the conjecture holds from the argument above. Therefore, the first interesting low dimensional case is \(K(5, 6)\).

The following section will describe the experiments we ran to find evidence to the conjecture 2.2.2. To test, we focused a lot on the \(K(5, 6)\) case, since it is the first case we did not know the behavior.
Chapter 3

Experiments

The goal of the experiments was to find evidence or a counterexample for conjecture 2.2.2. The main idea is to generate bipartite frameworks whose points are not strictly separated by a conic using the linear system described in the previous section, and then use theorem 1.2.3 to check its rigidity by looking for a maximum rank PSD equilibrium stress matrix of our framework. The tool used to find PSD equilibrium stress matrices is semidefinite programming, which we will describe in the following section.

3.1 Semidefinite Programming

A semidefinite program (SDP) is an optimization problem of the following form:
find $X$

to maximize $C \cdot X$

subject to $A_1 \cdot X = b_1$

\[ A_2 \cdot X = b_2 \]

\[ \vdots \]

\[ A_n \cdot X = b_n \]

$X \succeq 0$

(3.1)

where $X, C, A_1, A_2, \ldots, A_n$ are square symmetric matrices of equal dimensions, and we denote

$b = (b_1, \ldots, b_n)^T$. The dual program of the primal program above is the following optimization problem:

find $y = (y_1, \ldots, y_n)^T$

to minimize $b^T y$

subject to $y_1 A_1 + \cdots + y_n A_n - C \succeq 0$

(3.2)

The following lemma is the main property of semidefinite programs.

**Lemma 3.1.1.** The primal problem has a solution $X$ if and only if the dual problem has a solution $Y$.

This lemma is really important in our test cases as we will see in the following section. This is because in our SDP our dual problem finds an equilibrium stress matrix. And since our primal problem will always have a solution, we will always find an equilibrium stress matrix, and therefore we just need to test its rank to see if its maximum.

To run semidefinite programs we used MATLAB along with the VSDP software package, along with the solver in the SDPT3 package. The documentation of these packages can be found in [4] and [5]. These provide iterative numerical algorithms that solve semidefinite programs, and it looks for the highest rank solutions to the problem, which is exactly what we want. Of course, the algorithm gives us an approximation for the solution, but it will be good enough to accurately calculate the
3.2 Experiment setup

3.2.1 Generating Points

First we generate $N$ points in $\mathbb{R}^d$. In most of our tests, $N = (d + 1)(d + 2)/2 + 1$, which, by the separation criterion described in section 2.1, is the minimum number of points necessary to guarantee the partition into two sets strictly separated by a conic (assuming the points are not themselves in the same conic, as explained in the remark 2.1.3). In particular, for $d = 3$, which is the first unsolved case, we generate 11 points. The experiment was setup so we could generate these points in different ways, such as randomly chosen on an interval, points chosen on (or close to) a sphere or hyperboloid (in 3D), etc. Let the generated points be $P = (P_1, \ldots, P_N)$.

Then, we consider the image $P'$ of the points in $P$ under the Veranese map, so that each $P'_i$ is in $\mathbb{R}^{(d+1)(d+2)/2 - 1}$. We then use Radon theorem to separate those points into two sets whose convex hull in $\mathbb{R}^{(d+1)(d+2)/2 - 1}$ intersect. In order to do that, we setup the linear system

$$x_1 P'_1 + \cdots + x_N P'_N = 0$$

$$x_1 + \cdots + x_N = 0$$

We then take a look at the signs of each Radon coefficient $x_1, \ldots, x_N$, and let $\lambda_1, \ldots, \lambda_n$ be the nonnegative coefficients, and $-\mu_1, \ldots, -\mu_m$ be the negative coefficients. And we'll define $p$ as the points $P_i$ with nonnegative $x_i$, and $q$ as the points $Q_i$ with negative $x_i$. That is, the convex hulls of $p'$ and $q'$ intersect at the point

$$\lambda_1 P'_1 + \cdots + \lambda_n P'_n = \mu_1 Q'_1 + \cdots + \mu_m Q'_m.$$ 

Note that, on this process, we don’t choose the distribution of points in each set. That is, we don’t explicitly ask for a $K(5,6)$ rather than a $K(4,7)$. But to analyze specific distributions we just
keep generating cases until we get the desired distribution. In particular, for \( d = 3 \) and generating random points in the unit cube, the distribution \( K(5,6) \) appears about 83% of the time, so it is more likely than \( K(4,7) \). So since we know that the conjecture is valid for \( K(4,7) \) we just discard these cases.

### 3.2.2 SDP Inputs

After generating the two sets of points \( p \) and \( q \) for testing, we will setup an SDP to find a stress matrix for the graph. Using the notation similar to the one in equation 3.2, our parameters are the following:

- For each \( i = 1, \ldots, n \) and \( j = 1, \ldots, m \), we define \( A_{i,j} \) as the \( N \times N \) matrix such that the entries \((i, i)\) and \((n + j, n + j)\) are equal to 1, and \((i, n + j)\) and \((n + j, i)\) are equal to \(-1\). Note that we have \( mn \) of those matrices, one for each pair of points \( p_i \) and \( q_j \).

- For each \( i = 1, \ldots, n \) and \( j = 1, \ldots, m \), we define 

  \[ b_{i,j} = \|p_i - q_j\|^2 \]

  and denote \( b = (b_{1,1}, b_{1,2}, \ldots, b_{n,m})^T \) as a column vector.

- Since we only care about the existence of a PSD matrix, not the optimization, we will set \( C \) as the \( N \times N \) zero matrix.

Then we have our primal problem as

\[
\text{find } \quad X \\
\text{subject to } \quad A_{i,j} \cdot X = b_{i,j}, \quad \text{for every } i = 1, \ldots, n \text{ and } j = 1, \ldots, m \\
X \succeq 0
\]
and our dual problem

\[
\begin{align*}
\text{find} & \quad \mathbf{y} = (y_{1,1}, y_{1,2}, \ldots, y_{n,m})^T \\
\text{to minimize} & \quad \mathbf{b}^T \mathbf{y} \\
\text{subject to} & \quad \sum_{1 \leq i \leq n, 1 \leq j \leq m} y_{i,j} A_{i,j} \succeq 0
\end{align*}
\]  \quad (3.4)

Denote \( P = (p_1, \ldots, p_n, q_1, \ldots, q_m) \), considered as a \( d \times N \) matrix. Note that \( Y = P^T P \) satisfies the primal problem, since

\[
A_{i,j} \cdot (P^T P) = p_i \cdot p_i + q_j \cdot q_j - 2 p_i \cdot q_j = \|p_i - q_j\|^2 = b_{i,j}
\]

and, by the equivalent definitions of PSD matrices described in section 2, \( P^T P \) is PSD.

Therefore, by Lemma 3.1.1, we have that the dual problem also has a solution. Note that if the variables \( y_{i,j} \) in the dual problem are interpreted as stress for our framework \( (K(n, m), (p, q)) \), so that \( y_{i,j} \) is the stress for the edge \( p_i - q_j \), we have that the expression

\[
\sum_{i,j} y_{i,j} A_{i,j}
\]

is exactly the stress matrix related to the stress \( \mathbf{y} \), by the definition of the \( A_{i,j} \)s. Therefore, since the dual problem in 3.4 has a solution \( \mathbf{y} \), we have that the framework has a PSD stress matrix \( \Omega \).

Our strategy then is to use a SDP solver to find such matrix \( \Omega \) and check if it has the maximum rank \( N - d - 1 \). If it does, then the framework is universally rigid by theorem 1.2.3.

### 3.3 Test Cases and Results

The goal here was to exhaust the possible cases where we could find a counterexample for the conjecture. We first started by testing with randomly generated points. Then, we also focused on possible corner cases where we could possibly find a counterexample. Since the main condition is based on conic separability, one of our main focus was to test points that were generally close to a
conic.

Let $D = (d + 1)(d + 2)/2 + 1$. Our point generation varied as follows:

- Random points in the unit hypercube
- $D - 1$ random points on the unit sphere, $D$-th point on the origin
- $D$ random points ”close” to the unit sphere (added noise of order $10^{-4}$)
- $D - 1$ points in the unit sphere
- For $d = 3$, $D - 1 = 10$ points on the hyperboloid of one sheet

We exhaustively tested each of the point generation techniques for $d = 3, 4, 5$, and in all of the tests we had a PSD stress matrix of maximum rank.

Remark 3.3.1. We ran these experiments as an attempt to find a counterexample to the conjecture 2.2.2, or get evidence for its veracity. Later, in [2], this conjecture was proven to be true, as suggested by our results.

### 3.4 Tests on Incomplete Bipartite Graphs

After fully characterizing the conditions of rigidity on complete bipartite graphs, we attempted to also find some structure on the rigidity for incomplete bipartite graphs. That is, given a complete bipartite framework which we know to be rigid, by the conic non separability criterion, can we remove edges so that the framework remains rigid? If so, how many and which edges can be removed?

The first immediate condition for rigidity of a framework $(G, p)$ on $\mathbb{R}^d$ in general linear position (not only for the bipartite case) is that the degree of each vertex must be at least $d + 1$, from the argument in section 1.3.

Therefore, for the $d = 1$, $d = 2$, and the $d = 3$ case on $K(4, 7)$, we cannot take any edges, or we will have a vertex with degree else than $d + 1$. So, once again, the first interesting case is the $K(5, 6)$ on $\mathbb{R}^3$. 

26
We set up an experiment that starts by generating the points, just like in section 3.2.1. Then we remove each edge of the graph at a time and apply the rigidity test with SDP based on theorem 1.2.3 to check if the removed edge affects the rigidity of the framework. If an edge can be removed without affecting the rigidity of the framework, we’ll call the edge removable. To analyze the results, we plotted framework and colored the edges based on removability. We tested several cases on $K(5,6)$ for $d = 3$, and $K(8,8)$, $K(7,9)$, and $K(6,10)$ for $d = 4$. Some results are displayed in figures 3.1-3.5, where the removable edges are displayed in blue. In general, the $K(5,6)$ has an average of about 11 removable edges, but we found cases with as low as 6 removable edges, and as high as 15.

![Figure 3.1: Random rigid $K(5,6)$.](image)

We also tested the maximum number of edges one can remove at once from a graph. Because
Figure 3.2: Rigid $K(5, 6)$ with 10 points in the unit sphere, and one point in the origin. Note that in this case the 10 points in the sphere are non separable by a conic, so all the origin point (and consequently all of its edges) are removable.

This computation is rather complex, we only tested for $K(5, 6)$. In the vast majority of cases, the maximum number of edges that can be removed is 3, as in the one in figure 3.6. However, we also found cases of 2 and 4 as maximum number of edges that can be removed.

In summary, there’s no immediate conclusion on the rigidity of incomplete bipartite graphs that doesn’t depend on the geometry of the framework, since, as seen above, the measurements depend a lot on the particular cases tested.
Figure 3.3: Random rigid $K(8,8)$ in $\mathbb{R}^4$, projected to $\mathbb{R}^3$. 
Figure 3.4: Random rigid $K(7, 9)$ in $\mathbb{R}^4$, projected to $\mathbb{R}^3$. 
Figure 3.5: Random rigid $K(6 - 10)$ in $\mathbb{R}^4$, projected to $\mathbb{R}^3$. 
Figure 3.6: Rigid incomplete framework, of a $K(5, 6)$ with 3 edges removed.
Bibliography


